Mobile App Programming II Bibliography Part II

Matt Zagame

April 13, 2022

References

[10.1]   Kirkpatrick, Keith. Still waiting for self-driving cars. *Commun. ACM*, 65(4):12–14, mar 2022.

Abstract: The introduction of self-driving cars has garnered a large media presence over the last several years. While experts have been calling for public release of fully autonomous vehicles for a few years now, the reality is that safety concerns and regulatory issues remain largely in the way. Out of the 5 levels of autonomous driving capability, most new cars fall somewhere within the first three levels. As for the technology, it simply isn't there yet. The cameras and sensors that detect the world around the vehicle as well as the algorithms that act upon this data are still years away from being perfected. As a matter of fact, there are an indefinite number of variables that the AI of an autonomous vehicle needs to be able to handle which is no small task. The systems involved in training autonomous vehicles also require some degree of understanding of human behaviors. For instance, when a pedestrian is crossing the road, who may or may not be aware of the presence of the driver of the vehicle. Creating an autonomous vehicle that is aware of social cues is an intensive process that may take years and massive compute power to accomplish. In addition to the technology, one of the biggest challenges self-driving cars are facing is regulatory clarity. One cause for concern with self-driving vehicles is due to all the distractions built

into the vehicles that can lead to a driver not paying attention to the road.

[10.2]   Smallwood, Thomas R. C., Lefebvre, Véronique, and Bengtsson, Linus. Mobile phone usage data for disaster response. *Commun. ACM*, 65(4):40–41, mar 2022.

Abstract: Natural disasters can affect large populations and displace people from their homes. The act of mobilizing data to help identify locations of those affected can help humanitarian efforts during times of crisis. By aggregating Call Detail Records (CDRs) with the support of Mobile Network Operators (MNOs), it is possible to gain insight about the location of a population without breaching privacy. CDRs typically contain information such as the time of a call or text message, and the cell site the communication was routed through. Since the CDR data is collected in large quantities in near real-time, tracing the location of populations can be done rather quickly during a disaster. Due to being the most valuable indicators during a disaster, the process of disseminating indicators from CDR data remains to be an area for improvement.

[11.1]   Allen, Christopher. The path to self-sovereign identity. *Life With Alacrity*, apr 2016.

Abstract: The digital age has brought with it an endless amount of information sharing across web-based platforms that are responsible for keeping user information secure and, to some degree, private. However, given that user data is managed by services online instead of by the individual, a lot of personal information is often shared and even sold to other platforms or companies online. What if there was a way to have a trusted digital identity that also maintained user privacy? This is where the idea of a self-sovereign identity comes into play. Self-sovereign identity would allow anyone to establish and maintain their own digital identity, as opposed to being assigned an identity from a third-party platform or service. The path to self-sovereign identity is described to take place

over four phases: centralized identity, federated identity, user centric identity, and finally self-sovereign identity. In order to achieve truly self-sovereign identity, ten principles need to be incorporated, namely existence, control, access, transparency, persistence, portability, interoperability, consent, minimalization, and protection.

[11.2]   Denning, Peter J. Systems abstractions. *Commun. ACM*, 65(4):22–24, mar 2022.

Abstract: Abstraction in computational systems such as operating systems and more particularly computer processes are a strong foundation that has been around for decades. Originally, abstractions for operating systems were built upon modules and interfaces, which proved to be not all that efficient and difficult to debug. Thus concurrency control had to be established in order to handle all of the computations being run by many users. After processes were introduced into the operating system architecture, the idea for non-terminating computation using waiting states came about, leading to the modern design for operating systems that is used today. In a general sense, abstraction is an important aspect of any complex system. Abstractions typically occur in layers. Most modern operating systems use layered abstraction from lower level abstraction such as kernel operations, to higher level abstractions like user processes and services. The layered approach allows programmers to build on top of the lower levels without needing to directly interact with those layers. Overall, abstraction is quite a powerful tool in the study of computing.